A locating objects system using adhesive devices

TFG - Final Project

Abraham Cortes Vicent
COMPUTER SCIENCE UNIVERSITY OF BARCELONA 2015-2016
DIRECTOR: JUAN CLIMENT

Index

1. Introduction

1.1 Context

We live in a quick society without enough time and we go from one place to another using different means of transport. Under these circumstances it is very easy to lose objects which are relevant to us such as mobiles, keys, wallets, bike helmets, etc ... Therefore there is a need for technology in order not to lose any more important objects or at least, not lose them all.

The purpose of this work TFG (final grade work) developed at the Computing Science University of Barcelona is to create a adhesive device to any surface so that when the device is more than a few meters away from the mobile, it sends a signal to it. Not only that, but with a mobile application, it will be possible to set the distance in which you want to send the signal mentioned above and also may change the name of each device so that it is more recognizable by the user.

As explained in the following sections, in order to make this device, different phases will be taken such as the creation of the device, the setting up and programming of the hardware, software programming (mobile application) and others.

It will be our aim to make the size of the device as small as possible so that it will be easy to carry. Regarding costs, in order to create the prototype device, they will be quite high, but once created, its manufacturing and production will be considerably reduced.

Finally, it must be said that until quite recently, no similar device existed to the idea presented above. Currently, three French brothers have released a device which is used to prevent losing objects. As you will see in the following sections, although very similar, it is not exactly the same. In addition, the unit price of their product is much higher than we are aiming at with the TFG. The product is called Wistiki and their website is: www.wistiki.es.

1.2 Objective

The idea of this so-called TFG(Final Grade Work) System for locating objects using adhesive devices is to create chips that can stick to any surface except mobile, in order to help avoid losing any subject relevant to the person.

The functioning is simple and intuitive. When the phone is more than x meters away from the adhesive device, it sends a signal to warn that you may be losing an object.

The stages to be followed by users are:

- Stick the adhesive device on the desired object (except phone).
- Download the mobile app, which will be free for everyone.
- Log application and wait for it to connect the devices.
- Set each adhesives device, names and meters (desired distance).

Taking into account that each of the devices can be set at a different distance, that is, the keys, for example, can be set at one meter and the motorcycle helmet at 10.

One of the goals is to create a small size of adhesive device. To do so, an Arduino Nano with a processor module Atmega328 size 3.3 cm x 1.8 cm x 0.3 cm to 5V has been bought, also a Bluetooth LE Adafruit Bluefruit Friend UART and a speaker.

Finally, in order to make it a competitive product, it has to be efficient as to power consumption. For this reason, although different technologies with barely any consumption have been looked into such as NFC (Near Field Communication), we have decided to use Bluetooth low energy. Apart from being a technology that consumes very little, there is plenty of data. Moreover, NFC technology is not sufficiently developed, and its range is only 20 cm.

1.3 Actors Involved

To make this project come true, there are different actors involved. First we find the developer. He/she is responsible for research, design, manufacture, and programming required for the mobile application and hardware. Also, if necessary, this actor would as well be responsible for marketing the product.

Being a TFG, the director is also involved. Their responsibility is to try or rather guide their student in order to do a good job and also help them in what they require (making the product is not their responsibility). Regarding this same aspect, there are the tutors from the GEP subject, which is compulsory once you join the TFG. They also guide you and help you in the development of TFG and give answers to submissions that were made so that the work presented is at least appropriate to the required level.

Finally there is the final user. The compelling of this project is that it covers a great quantity as well as different type of people. While, at the beginning, this project only aimed at the Catalan society and perhaps the Spanish, it could easily be marketed worldwide.

1.4 Development

Different issues have been taken into account in order to create this chip. The most important is the communication between the chip and the mobile device. In this case we want to use Bluetooth low energy, Beacon in particular, for different reasons.
First of all, Bluetooth low energy can be used in almost all current mobile phones, thus avoiding compatibility problems. Secondly, this significantly reduces the consumption of the standard Bluetooth. Thirdly, if we use beacon, we need to do it with the Bluetooth low energy. Finally, it must be said that there is a lot of data about mobile applications that use Bluetooth low energy with Beacon.

	Classic Bluetooth technology	Bluetooth low energy technology
Data payload throughput (net)	2 Mbps	~100 kbps
Robustness	Strong	Strong
Range	Up to 1000m	Up to 250m
Local system density	Strong	Strong
Large scale network	Weak	Good
Low latency	Strong	Strong
Connection set-up speed	Weak	Strong
Power consumption	Good	Very strong
Cost	Good	Strong

Figure 1: Comparison BLE vs B

Beacon is simply a device that is sending the information broadcast by UUID, Major and Minor. Besides, it sends its Mac, the device name and information of the GATT services that will be explained later.
In Section 3, it is thoroughly explains the development of the system for locating objects using adhesive devices, including the Software and Hardware.
In short, the phases that have been followed by its creation are:
- Choice of Hardware
- Setting, development and installation of Hardware
- Power consumption
- Welding of the Hardware
- Issues raised when developing Hardware
- Choosing Software
- Development of mobile application
- Issues raised when creating the mobile application
- General issues raised between Software and Hardware .

1.5 Working methods

The methodology to be used in the TFG is Scrum. This method is the most suitable because of the flexibility it offers. It works as follows:
First the roles ScrumMaster, ProductOwner and Development Team are defined.

- ScrumMaster controls and guides to make the practice or project forward successfully.
- ProductOwner represents the stakeholders in the project
- Developing Team are all those involved in software

In our case, all the three roles are carried out by the same person. After defining roles, you need meetings, usually every morning to see how it progresses and then modify it according to forecasts.
For this reason, this methodology is so suitable for this work, because if you come across any problems or drawbacks, it allows you to modify it.

2. Market Research

As said in the introduction it was until recently that there was no similar device in Spain, now we currently found the so called WISTIKI.

2.1 History

This project began two years ago in France in the National Center for Scientific Research laboratories in Lyon city. They are three brothers who founded the company in January 2014.
Before the merchandizing, in November 2014, there were 35,000 Wistikies ordered by different production chains, and by the end of 2014 there were over 50,000.

Figure 2: creators of Wistiki

2.2 Features

In this section we will highlight the most important features of this device.

2.2.1 Functionality

This device has four functions:

- Bell & Radar
- GPS Position
- Reverse Bell
- Virtual Strap

Figure 3: Wistiki

2.2.1.1 Bell & Radar

This is the main functionality, which is used to find lost object the sound of the Wistiki. It also has a radar so you can play the legendary game hot or cold.

2.2.1.2 GPS position

As the title suggests, this is used to find the lost object through the mobile GPS.

2.2.1.3 Reverse Bell

If by any chance you lose your mobile, this functionality will recover it. Using any mobile Wistiki will make it ring.

2.2.1.4 Virtual Strap

You can activate this functionality to alert you when the phone is away from the smartphone.

2.2.2 Autonomy

The Wistiki requires a Maxell CR2032 battery with a capacity of 220mAh (recommended) , which has a range of one year.

Figure 4: MAXELL CR2032battery

2.2.3 Scope

The GPS has no limitations when available, but Bluetooth is 50 meters maximum

3. Development of the system for locating objects using adhesive devices.

As mentioned above, in order to develop the TFG (technical part). phases or stages have been specified as follows:

- Choice of Hardware
- Setting, development and installation of Hardware
- Power consumption
- Hardware Welding
- Issues raised when developing Hardware
- Choosing Software
- Development of mobile application
- Issues raised when creating the mobile application
- General issues raised between Software and Hardware .

3.1 Hardware Choices

It must be taken into account that in a project that includes both hardware and software, this phase is very important because depending on the hardware chosen, the outcome can vary greatly and can make the project work or not. So I had to perform this stage focusing on different factors.

The first and most important factor is the consumption. But we will deal with it in Chapter 4. Another important factor is the size of the project and therefore its weight. And last but not least the price.

The components selected by the factors mentioned above are as follows:

- Arduino Nano (Atmega328 processor)
- Adafruit Bluefruit LE UART Friend (Bluetooth Low Energy)
- Loudspeaker or Small Speaker

We also need different tools to work with as:

- MicroUSB to USB cable for connecting the computer with Arduino Nano
- Breadboard (To make the programming before welding)
- Male Cables - Male (For the Breadboard)
- Philips 9V battery with the connection head
- Welder with pond.

- Arduino IDE Software (to able the setting and the programming)

The Arduino Nano has the following characteristics:

- 14 digital inputs and outputs
- 8 analog inputs
- 16MHz Clock
- Dimensions: 3.3 cm x 1.8 cm x 0.3 cm
- Weight 0'005 kg
- Price 5'87 €

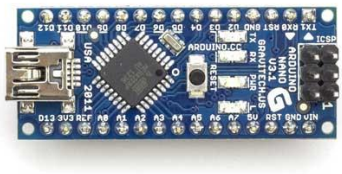


Figure 5: Arduino Nano Atmega328 processor

This device has been chosen mainly because of its size and price. Another factor, which is not among the most important, but has been chosen because of its relevance, is the fact that Arduino has been extensively tested and proven. It also has a huge community behind it that has found many bugs and problems as well as many solutions. This fact is important because when purchasing it, you also purchase reliability.

Another reason for this choice is due to the features it includes. This device is the core hardware that means will control and feed the other components. It accepts an input voltage between 7 and 12 volts which is perfect for 9V battery. At least pin serial communication will be needed for the Bluetooth. The Arduino Nano, apart from having a serial communication plate, it gives you the opportunity to use the library SoftwareSerial (which allows digital pins operate as serial communication pins) on all its digital pins. We also have output for different Grounds and 5V and 3v3, to feed components according to its need. Finally, it has also been chosen as the speaker will need a PW pin M and the Arduino has got 6. (In the appendix there is the Arduino Nano datasheet where you can see all these data).

Using another type of microntolador, such as Raspberry Pi zero, was put forward. This device has the following features:

- 1GHz, single cpu -Core
- 512MB

- HAT-compatible 40-pin header
- Micro USB power
- Input voltage: 5V
- Dimensions: 65 mm × 30 mm × 5 mm
- Price: $ 5

Although it is a microcontroller more powerful than the one chosen, it was ruled out mainly due to its size, which almost doubles the Arduino Nano. In addition, its input voltage is 5V just like the Arduino Nano (although re7V to 12V is commended). Another reason is that this project does not need a 1GHz processor, this usually entails more consumption.
Another microcontroller that was considered was SmartEverything. This has many more options, such as GPS, Sigfox, accelerometer, etc .. And precisely for this reason it was also ruled out. It is a too big hardware for this project. Moreover, the price is around € 100 and it must be said that the community information is null or virtually nonexistent. Finally, it was also ruled out because of its size (53mm X 68mm).

Adafruit Bluefruit LE Friend UART has the following features:

- ARM Cortex M0 core running at 16MHz
- 256KB flash memory
- 32KB SRAM
- Transport: UART @ 9600 baud
- On-board 3.3V voltage regulation
- Dimensions: 21mm x 32mm x 5mm
- Weight: 3.4g
- Price: 15 €

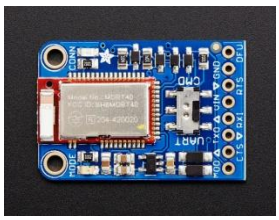


Figure 6: Adafruit Bluefruit LE UART Friend

This Bluetooth has been chosen for several reasons. The first and most important reason is that it can act as a beacon, which is essential for this project. Also because you can create personalized GATT features and have access to them. Another reason is the fact that the community is behind and that includes tutorials.

As it can easily be seen in the features, its weight is ridiculous and the dimensions match what we want the final product to be. It needs a 3v3 voltage which the Arduino Nano 3v3 provides. The clock is 16MHz like the Arduino , although as it works by serial communication, it is not a problem. The only drawback is the price, as € 15 is too much for the work it would do.

Initially, this Bluetooth was not our choice, the one that was really wanted was the HC-05 Bluetooth low energy module. Although it required welding, it is smaller and the price is also lower. The problem was that practically no information was found, all the features were on the same sales page and no datasheet or other useful information, such as AT commands or the option to add GATT features, was found . For this reason Adafruit Bluefruit LE Friend UART was finally chosen.

To conclude, the Speaker is an indispensable part of this project. In the end we opted to have a small and simple speaker. This speaker can be found at any computer tower or can be purchased for € 0'20.

Figure 7: Speaker

As first tools for the proper functioning of hardware we have the USB cable – micro USB, the breadboard and male wires for the breadboard connections.

The USB - microUSB, will be used for the connection between the computer and the Arduino Nano and has a size of 29cm from end to end. The breadboard has 650 points with two connections both positive and negative and dimensions of 16'7cm long and finally the male cables - male 6'5cm long.

One of the alternatives to all this hardware would have been the BLEduino. This device is the size of Arduino Nano and it also incorporates Bluetooth low energy. The problem is that it is still in developing phase and since 2014 nothing has been said anew. If it was

finally launched this project would be reconsidered to make it work with this device.

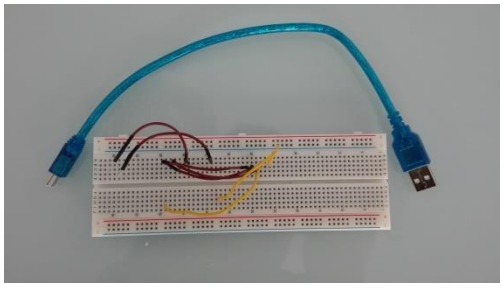


Figure 8: BreadBoard and wire

Philips battery is 9V. For Arduino to work properly you need en entry between 7 and 12 V. This one has cost 3'50 €.

As for the welder and the tin, a kit was bought with everything included (welding iron, unwelding, tin and others) for € 20.

Finally, you also need the Arduino software. This program allows you to set and program the Arduino Nano. This program was chosen, above all, because we were already familiarized with it. We must also say that it is a free program designed by the same company that makes the Arduino Nano. Conclusively, we would like to say that one of the great advantages of using this program is the great community and the years that has been running. As mentioned before, not only you purchase a great program but you also purchase reliability.

3.2 Configuration, development and installation of Hardware

3.2.1 Setting up

The first step is to set up the hardware. In the figure below you can see the connections between Arduino Nano and Bluetooth low energy (Figure 9).
Adafruit Arduino Nano Bluefruit LE UART Friend

Arduino Nano	Adafruit Bluefruit LE UART Friend
GND	GND
5V	VIN
D11	CTS
D10	TX0
D9	RX0

Figure 9: breadboard connections table

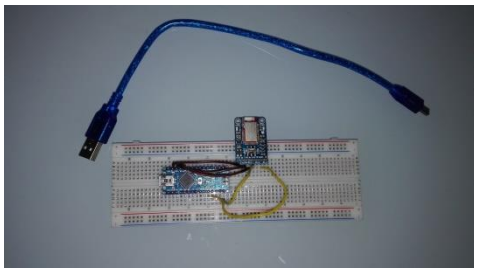


Figure 10: Connections between Arduino Nano and Bluetooth low energy

The CTS pin is used to indicate the same Bluetooth that it can send data via pin TX0. The D10 and D11 pin is used for serial communication between Arduino Nano via Bluetooth and Serial Software bookstore and finally the Bluetooth GND pin to the Arduino GND and the Vin pin is connected to the Arduino 3v3.

As for the speaker, it has been connected to the positive (yellow in Figure 11) to pin D6 acting as PWM and GND to GND.

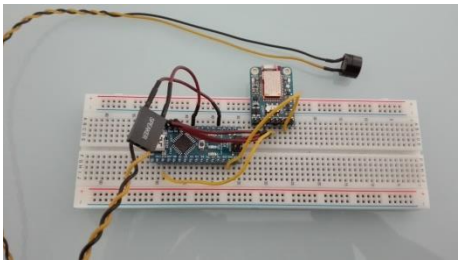


Figure 11: connections between all components

3.2.2 Hardware Configuration

In order to understand settings that are being made, we must first know what the desired operation by the hardware is.

What we want is to create a GATT service with a feature that can read and write from any Bluetooth device that knows the identifier of the feature. The hardware will be constantly reading this feature which can be 0 or 1.

In case it is 1, it will send a signal to the speaker to start ringing. For 30 seconds it will continue ringing unless the feature becomes 0, when it will stop ringing and will wait again until the value is back to 1.

The following figure shows, in a diagrammatic form, the device firmware:

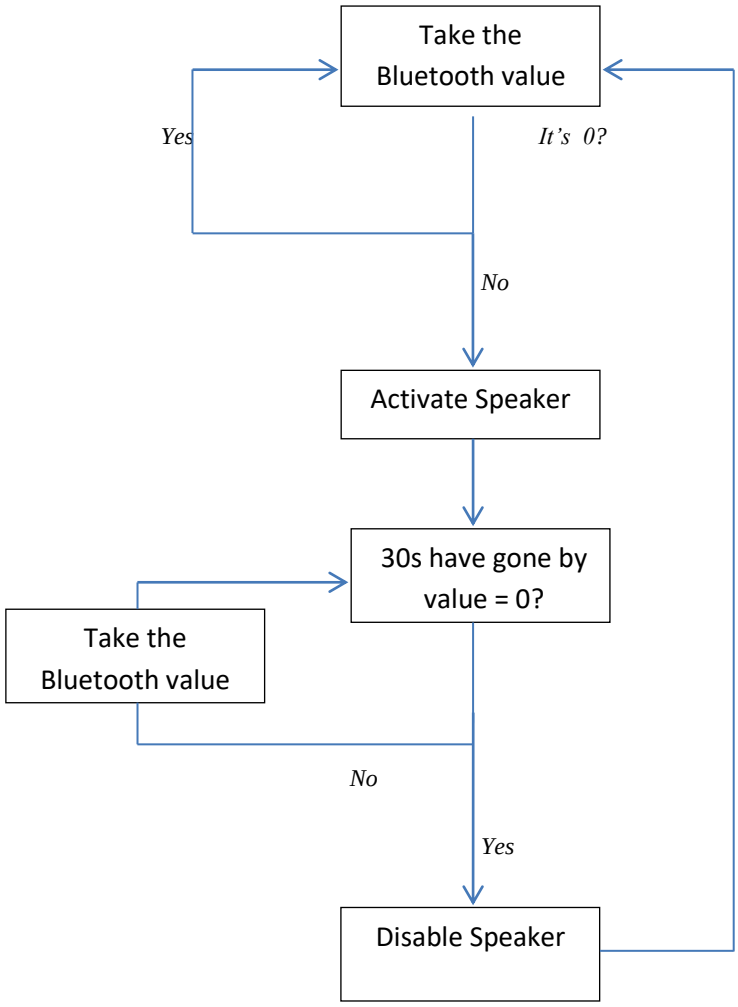


Figure 12: Flowchart

The Arduino Nano does not need any special configuration in order to work but you need it to configure the Bluetooth and speaker. All connections described in the previous section indicate the Arduino. For the speaker we put the D6 pin as output with a frequency that we decide. It must be said that all the pins of the Arduino Nano 3v3 operate in volts, which are necessary for the speaker and the other components. As for the bluetooth the bookstore will be called and get Software Serial pin D10 as TX (transmission) and D9 pin as RX (reception).

For the Bluetooth settings (which are made by Arduino Nano and serial communication) the following steps have been followed:

Firstly, we must make sure that Bluetooth is in CMD mode. This mode allows us to enter its configuration and change it via AT commands. Each device has a particular or different AT commands and they are dependent on the firmware version of the device itself.

AT command set used for Bluetooth as Beacon:

- AT & BLEBEACON = 0x004C, 01-12-23-34-45-56-67-78-89-9A-AB-BC-CD-DE-EF-F0,0x0000,0x0000, -59.

Being 0x004C, the Manufacturer ID indicating the protocol to follow. In this case it comes as iBeacon meaning that the Apple device will be able to find the 01-12-23-34-45-56-67-78-89-9A-AB-BC-CD-DE-EF-F0 uUID is identified as the beacon, the Major 0x0000 and the other minor and finally -59 is the RSSI (Received Signal Strength Indicator) to 1m.

The beacons are one-way communication only. Therefore in order to communicate with Bluetooth, we add features GATT in which we can read or write a value.

AT command used to add a personalized service to Bluetooth:

- AT GATTADDSERVICE = + = UUID128 00-11-22-33-44-55-66-77-88-99-AA-BB-CC-DD-EE-FF

This Command is used to add a service UUID 128 bytes and this is 00-11-22-33-44-55-66-77-88-99-AA-BB-CC-DD-EE-FF. This service is free which means you can add features. That will help us to change values and read them from the Arduino Nano and modify them.

AT command feature to add and modify the value:

- AT & GATTADDCHAR 0x2F30 = UUID =, = 0x08 PROPERTIES, MIN_LEN = 1, 0 = VALUE
- AT & GATTCHAR = 1.0

-AT+GATTCHAR=1

. Once we have the added service, a feature must be added with the corresponding UUID and its properties. This is what makes the GATTADDCHAR command where UUID is 0x2F30 and its properties are 0x08 (write unanswered), minimum length 1 and

the initial value 0. Finally we indicate the Bluetooth that we want this feature to have the value 0 and therefore we use the command GATTCHAR = 1.0 where 1 is the number of the feature and 0 is the value we want. Lastly we have the GATTCHAR = 1 + AT command, where the current value of feature 1 will be returned.

Command for advertising Data:

- AT & GAPSETADVDATA = 02-01-06-11-06-DD-EE-FF-AA-BB-CC-99-88-77-66-55-44-33-22-11-00-03-02 -0A-18

Finally, the GAPSETADVDATA command is used to modify the same Bluetooth "advertising data", thus indicating the available services so that they could be discovered. This is necessary as we have added a new service and asked the Bluetooth to display it in the "advertising data".

Once the Bluetooth is working as Beacon with the added feature, we will proceed to program its "normal" behavior. To do so, the main starts reading the feature mentioned and if it changes its value, then the speaker will be activated so that it will ring accordingly. For the time being, it has been established that as long as there is no other change (e.g. the feature goes back to 0 again) the buzzer will sound for 30 seconds, so you can find the object. This process can be repeated as many times as you want.

3.3 Energy Consumption

Consumption has evolved during the project, which is why in this section explains the steps that have been taking.

One of the main objectives was the battery consumption of the device to be as little as possible. The battery in this case is a PHILIPS 6LR61 9V battery that gives up to 600 mAh. The device has a consumption of about 33 mA when the value of the GATT feature is being checked, if the speaker is working then the consume reaches 92 mA. Assuming that it will ring once a day at the most, then the consumption average would be 33.07mA. So if we apply the following formula: (Battery capacity / consumption device) * (0.9) 1
(600mAh / 33.07mA) * 0.9 = 16:33 pm

The 0.9 is due to external factors that can affect the battery.

It is clearly one of the aspects of this project to be improved, because we cannot expect the user to change the battery every 16h. One way to improve it would be to make more the Arduino Nano a more efficient code Arduino Nano or reducing the battery voltage.

In this case the battery has been changed to a CEGASA CR-P2 6V which gives up to1600mA . Testing the tester can show that the consumption decreases when getting to 26mA and 84 mA when no sound is ringing. Assuming exactly the same as in the previous case then we get that the average consumption is 26.02mA. Then using the

formula described above we obtain:

(1600mAh / 26mA) * 0.9 = 55.38 h

Thus obtaining an improvement (55.38 / 16.33) * 100 = 339%. Although it is a considerable improvement, it means that the user should change the battery about every two days. It is still insufficient.

At present, we have tweaked the code to make it more efficient as far as consumption is concern, setting the device in sleep mode for 1 second after reading the data. It has also been applied that instead of the speaker ringing for 30 seconds nonstop, it will now ring intermittently during those 30 seconds. This has reduced the consumption 12mA when it is not ringing and 32 mA when it is not. So the average consumption becomes 12.1 mA. Applying the formula to know how long the battery is going to last, you get:

(1600mAh / 1.12) = 0.9 * 119.90h

The improvement obtained in this case is (119.90 / 55.38) * 100 = 216.5%, which means that the battery would last almost four days.
It is clear that consumption is still too high, so now we will try to spend two seconds in sleep mode instead of one, as we believe that it is not critical for the user to wait as long as two seconds for the sound to start. The consumption is then 8.2mA when it is not ringing and 32 mA when it is. The average consumption is then 8.21mA, so the battery life will be approximately:

(1600mAh / 8.21mA) * 0.9 = 175.40h

The result is an improvement of (175.4 / 119.9) * 100 = 146%. The battery will last for about seven days.

More tests have been done in order to better battery life but the results have not been satisfactory. We tried to reduce the consumption of the Bluetooth transmission by making the serial port transmission slower, but this directly affected the RSSI (Received Signal Strength Indicator) and this seriously harmed when calculating the distance. Bluetooth advertising data was also reduced, meaning that it send data that was less frequently discovered, but the improvement in consumption was negligible.

In conclusion it is clear that a 7 days is not bad for a prototype but if the product was to be sold, then it is totally inadequate. The problem is the consumption of the

microcontroller itself. If the product was to be launch to the market it would need to be changed into a low voltage series in order to try to improve consumption.

It must be said that seeing that the final consumption was not the one desired, a turn off button has been added to the design of the device. This will make the battery last longer and give the user the option to comfortably disconnect the device.

3.4 Problems during the course of hardware development

The phases or stages discussed above have taken longer than initially expected. This delay in the estimation of deadlines has led to a number of different problems during the development phase.

One of the greatest problems was that the Bluetooth could not be set up correctly. It could not be set in command mode, and it would not respond to any AT order. After searching on the Internet and trying different experiments, as trying to write the AT commands through serial port manually, we found out that there was a conflict between the SoftwareSerial libraries and the libraries of the Bluetooth. In order to solve this problem a mobile application of the Bluetooth manufacturer was installed and that gave access and the firmware could finally be updated. Once the updated was done, using the SoftwareSerial library, there was access to the Bluetooth setting mode and put the AT discussed above.

A further problem was that sometimes the serial communication failed, and as a result some settings introduced were not posted correctly to the Bluetooth. The solution, in this case, was quite easy as a loop for each of the commands was made where, until the Bluetooth would not answer "OK" through the port serial, it would keep sending the order.

3.5 welding Hardware

To avoid having to use the breadboard further, once the hardware already works as desired, this will be weld as in the following scheme:

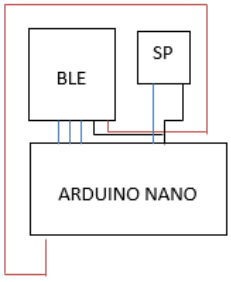


Figure 13: connections arduino

As you can see in the figure it was welded as it was in the breadboard. It must be said that it was welded on a connections plate and then it was cut. The result can be seen in Figure 14.

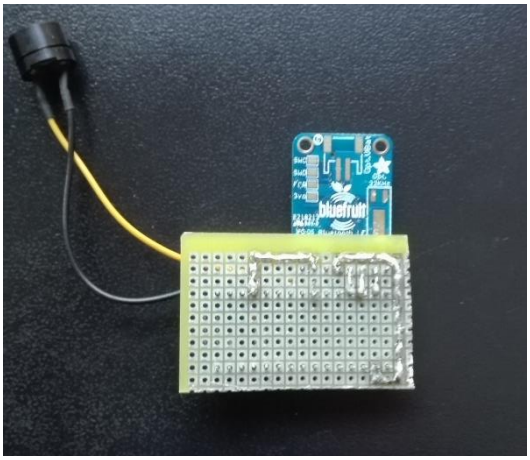


Figure 14: welding components

Finally, it was also decided to design a 3D box that would be a container for device and battery. To design it, the trial version of the Sketch Up software has been used. After that, a plugin has been used to export the design in a STL format because of the 3D printer needs. The design is as follows:

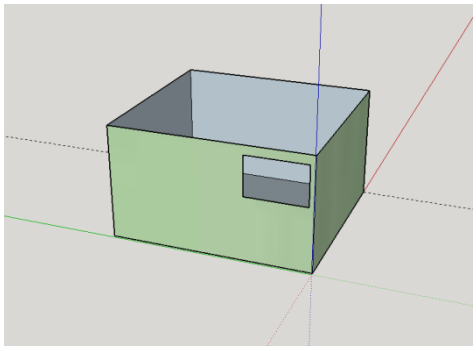


Figure 15: Design box

As you can see there's a hole in the box. In it, there will be a two-position switch to turn the device on and off the, thus improving power consumption.

3.6 Choosing Software

To successfully complete the mobile application some software is needed to develop it. In this case we have chosen Android Studio.

This is free software that allows Android application development. Although it is fairly new software, it has been highly expected by the android community. The most common software before this was the eclipse. It offers many options for programming and it is also designed and solely dedicated to the development of applications. It makes programming and especially the design of the app much easier. Finally we chose this software because we can find many tutorials and documentation on how to carry out android applications.

3.7 Mobile Application Development

From the start, it must be said that at the beginning of the TFG, we had no notion or previous studies about the development of Android applications. This has caused, among other things that the time spent on making the application increased considerably.

The mobile application is designed so that it can link a new device, edit the connected devices to change the name and the distance in cm to be set up and there will also be a section to observe linked objects and if needed make them ring. On the other hand you must run a service that works in the background, which goes on automatically analyzing the distance from devices. If one of these exceeds the distance scheduled, it will send a notice to the mobile indicating the name of the device if you want it to start ringing. If the user says yes, then the device begins to ring and a new screen appears that asks whether or not the object has been found. If the device is found, then it will stop ringing. If it is not, then it goes on ringing for 30 seconds.

First of all, it is important to explain just as the application has been thought; it is necessary a local data base, where the name of the device, the programmed distance for it to ring and finally the Mac address of the device are stored. This way we can manage all the operations required.

The application consists of different screens organized in menus. In the main menu you can find three buttons.
- Observing Objects
- Editing Device
- Linking Device

Figure 16: screen android application

As you can see the buttons are centered and each leads to a different Activity. We must also say that the service that works in background is executed in the main and it will examine each of the connected devices as well as their distance.

You will find a list of connected devices, and the distance at which they are on the Observing objects screen. In order to make this screen, a List View element has been added to the layout, which is filled by a function which observes the nearby beacons and checks whether they are linked or not, what RSSI (Received Signal Strength indication) they have in order to know distance and finally, it also checks the name of the device.

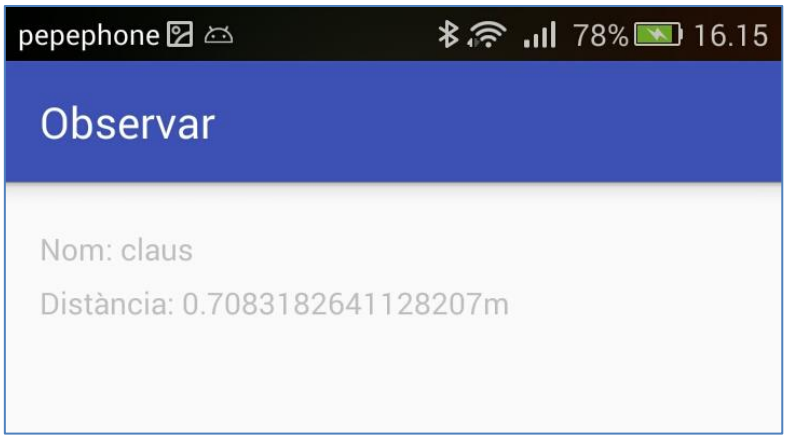


Figure 17: Screen Watch

In this case we can only see a device which we have kept by the name of keys. If we did not have any connected device, no item would appear. We can also see that it is at a distance of 0.7 m, ie 70cm.
In order to do this when creating the activity, the ScanLEDevice function is called. What it does is tell the Bluetooth to start scanning all Bluetooth devices. When anyone is found, the information turns to the mLeScanCallback function, which we have as "Device, RSSI, ScanRecord" input parameters. ". We only want to see connected

devices, and for this reason we do a SQL Select query on the local database, which will return the MAC address of the device. If the query result is smaller than 1, then we will ignore this device. If it is greater than 1, we create a list of two parameters where we keep the Mac device and its features GATT. Once we have this list, we make a ListTextView to put it on screen, where the name of the device and its distance is included.

The distance is calculated by the RSSI (Received Signal Strength Indicator), using a formula

Figure 18: Dialog Alert starts ringing

As we can see, when touching an item in the list we will get an "Alert Dialog" which asks us if we want the chosen device to start ringing. At the push OK, the device will start ringing for 30 seconds.

Having saved the MAC and features GATT, when pressing OK what we are doing is connecting the device and writing 1 on feature GATT (explained in the section on development Hardware). Then when the Arduino Nano detects this 1, it adds 200 in pin 6 (speaker pin). Finally, it disconnects. The disconnection is very important because if for whatever reason it remained connected, when the scanning was on, the device would not be found again or we could not discover any other devices. We could not even see the distance.

For safety and convenient reasons, it has been established that this process can be canceled simply by pressing the list item again. In this case, another "Alert Dialog" would appear which would ask us if we want device to stop ringing. As shown in Figure 19.

Once more, we will follow the same process, connecting through GATT features, write a 0 on the chosen feature and disconnect.

Figure 19: Dialog Alert Stops Ringing.

Another list appears on the Edit Device screen, but in this case, only of the linked beacons. It simply picks the linked beacons from a database, which has been created to control all the devices that one may have, and puts them on the list. The name of the beacon appears as, for example, the wallet and the distance that we have planned for it to ring, say 100 cm (Figure 20).

The ListTextView parameter is used again to make the list, but this time instead of looking out the devices that come from the Bluetooth, what is done is to fill it with the local database we have.

Figure 20: Edit Listing

When pressing any item of the list, we access to its configuration (Figure 22). In this configuration we can change both the name and the scheduled distance. As in the case of the wallet, could insert “keys”, 20 cm. When going back or accepting, we would see that we no longer have the “wallet”, but the “keys” to 20 cm (Figure 21).

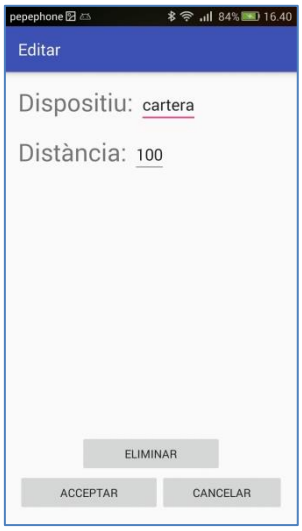


Figure 22: Specific Edit Screen

Figure 21: Edit Completed.

It should be said that when we press an element from the list, we enter a new activity, but this time we add the name of device and the distance using the Extras. In order to go from one activity to another, it is necessary to create Intent, and in this Intent, you can add the Extras which have two fields. One of them is the name you want to give it, and the other the value. So when we go to another activity, first you need to find the assigned name and retrieve the value.

Once the user presses “accept” or “OK”, we take the text from the TextView disp and the one from the TextView dist and update the database, the fields and the name and distance of the corresponding row.

We can also remove a device when, for example, we have given it away or do not have it for whatever reason. In this case it would follow the procedure below. First of all we would select it from the list and then click on “delete” where another "Alert Dialog" would appear and ask us whether we are sure that we want to remove it (Figure 23). If we “YES”it would then disappear from the database (Figure 24).

In this case, the procedure is exactly the same as in updating, but instead of making a SQL UPDATE statement we carry out a DELETE.

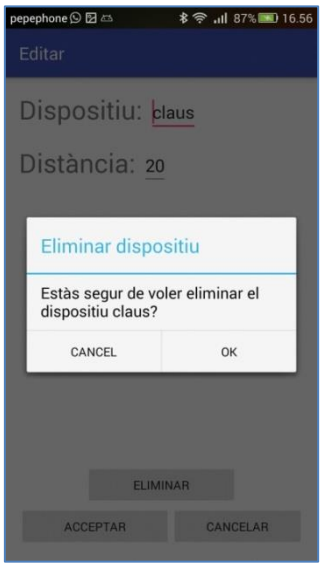


Figure 23: Dialog Remove Alert

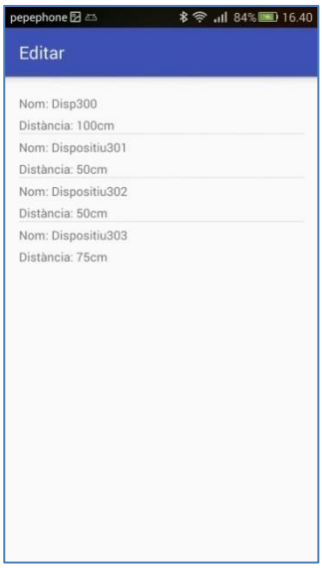


Figure 24: Edit screen Update.

Finally we have the Link screen devices. Like in the others, a list of beacons would appear, but in this case it would not be those that are linked but the rest (Figure 25). If we click any item in the list, we will receive a message that says if we are sure that we want to link this device (Figure 26).

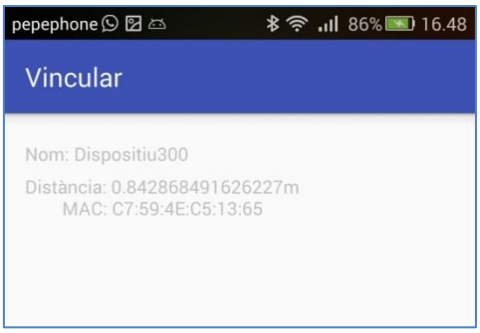


Figure 25: Linked Screen with List of Devices.

Figure 26: Dialog Alert Link.

When the device is connected, it means it is added to the database; therefore we can observe the object and make it ring if we wish or change the name and the distance for what may seem appropriate.

This Activity is very similar to that of observing, but in this case there has to be a filtering for the devices already linked not to appear. When a mLeScanCallback function returns a device, we take its MAC address and we make a SQL Select query to see if we already have this Mac linked. If the result of this query is greater than 1, then all you have to do is not to add this device to the list.

The operation and development of the service that works in background still has to be explained. The service, as mentioned above, is executed when the application starts. But in case this should be switched off, it would still be running. What this service does, as in observing or linking, is to use the Bluetooth and make it scan all the time. Each device that is found, it has to be checked in the database. If so, then we check the

scheduled distance (which is done using a SQL Select) and check the current distance of the device (by using RSSI). If the RSSI distance is greater than the scheduled, it sends a notification to the user giving the name of the device and two buttons to indicate whether or not you want the device to start ringing. (figura19). In order to carry out the notification, the Notification Manager library must be used. The icons of the notification, the title, the inner text, the actions (in this case the buttons) and finally an identifier for this notification must be specified.

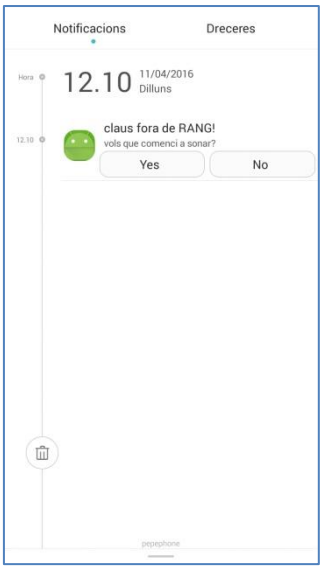


Figure 27: Notification out of range

If we press “NO”, then the notification is cancelled by the Notification Manager Cancel (notification ID). If we press “YES·, then the notification is cancelled, the device starts ringing and a new activity opens up (Figure 28), which asks whether or not the device has been found. For the device to ring, we do exactly the same as in Observing, which is to connect to Bluetooth, read the GATT features and write 1 to the desired feature. While it is ringing, the device appears on the screen, previously discussed above, with the “yes or no” buttons. In this case, if we press “YES”, meaning we found the device, then you reconnect the Bluetooth and write a 0 to the GATT feature, thus making the device stop ringing and closing the activity. On the contrary, if we press “NO”, the device would ring again. This process can be repeated as many times as you wish.

Figure 28: Alarm Activity

The process that follows when we press "YES "to the notification is not trivial. It was necessary to create a Broadcast Receiver which receives the response of the notification because this one, coming from a service and not the application, cannot take any context and therefore, we need this intermediate process that is actually checking all that is going on in the mobile. This Broadcast Receiver checks whether the answer was one or the other and, according to it, creates a new activity and cancels the notification or simply cancels the notifications. The activity Alarm calls the functions from the Service already created as to avoid creating unnecessary code.

It is important to mention that the initial duration of this section was 4 weeks. Due to unlearned programming language like the errors as explained below, the final duration was 12 weeks.

3.8 Problems during the creation of the mobile application

One of the main mistakes which made the duration greatly increase was that of the compatibility. The cell in which the tests were carried out was an API 19. This means that there are many functions related to Bluetooth low energy that cannot be used. . For example: the BluetoothLeScanner requiring API 21. Actually, none of the library Android.Bluetooth.le functions could be used. After a time of research online I found BluetoothAdapter.LeScanCallback API library 18. This library is fallen in disuse, but it is the only one that worked with the mobile.

Another serious problem was the usage of a database. Even though the SQL language was not an issue as I have deep knowledge, its creation through android was really complicated due to the fact that in order to create database, two activities must be created separately: one to create the structure and the other to create the functions that this SQL manages. In the end I got help from a library called SQLiteDatabase.

There was also an error with the Service. Once the entire system was operating, in testing I realized this service was not working in Background. This meant that if the application was not in the foreground or the mobile screen was at rest, the service stopped working. Consequently, the project would not make sense because it would force the user to always have the same application on the phone and not put the screen to sleep. In order to fix this, I searched on the internet and looked it up at the official Android Studio site. There I ratified that instead of declaring an activity as a Service, I had to declare it as IntentService. Although the change was not great, there were still things that had to change.

Other problems or errors were because of my lack of Android knowledge. This meant that I had to make the most of online tutorials than thought necessary.

3.9 General Issues between Hardware and Software

One of the most serious problems between hardware and software was sending the desired feature to the GATT service. At the beginning, when sending a 1 to the feature, it was perceived that Arduino Nano was not receiving anything, or what was received was not a 1. After searching on the internet and especially after seeing the working of the application of Adafruit, I realized that Arduino Nano, expected an Unsigned byte and the default application sent an array of bytes. In the end I could change this array of bytes for a uint8_t.

4. Future Project

As for the project itself, it has many possible modifications and upgrades:
- GPS locator
- Change Bluetooth for NFC (Near Field Communication)
- Improve Android application
- Thinking of business

GPS locator will be useful to indicate the exact position of the lost object with a simple mobile application or using Google Maps.

There are different issues. The design we want has to be as small and efficient as possible in power consumption. A GPS is usually of very large dimensions and consumption is high.

Relevant characteristics:
- Dimensions: 23mm x 35mm x 8mm
- Consumption (MTK3339): 25mA tracking, 20mA in navigation
- Supply: 3.0 - 5.5VDC

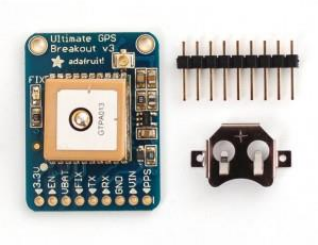


Figure 29: Adafruit GPS MTK3339

The Bluetooth low energy is currently a very efficient technology for energy consumption, but it is impossible to communicate with a device without electricity. Nevertheless, with NFC technology it is feasible. The problem is that at present this technology only has a 20cm range, Therefore, as it is right now, it would not be possible to use this technology.

Clearly, android application has many possible modifications and upgrades, among other things, due to my lack of sufficient knowledge. For example: the application consumption is very high because in order for this project to work, it has to be on permanently. It could be improved when even though the application was closed (killing the process that runs it), the service detection and alarm would continue to operate. Other areas for improvement would mainly be an aesthetic issue in order to make it more pleasing to the eye.

Another aspect liable to improvement or redoing would be to create a server where all the information of the manufactured beacons was stored. This way the company would be in control.

Other future challenges would not selling the device only to individuals but also to companies. It should provide a tracking system for all its devices through a website so that it could be interesting for companies. Consequently it would also need the website and servers required to be created. It would also be necessary for the device to send a signal either by same mobile or include sigfox phone or other communication systems.

5. Timing

In order to carry out the whole project, it is estimated that the time needed is of about four months. In the next section it will be explained why these four months, although the time may be higher due to unforeseen drawbacks which were not calculated. For this reason it is worth mentioning that this section is not rigid or immutable, other than an approximate, but accurate guide trying to define the necessary time and complications that may arise.

5.1 Stages of the project

The next data and project development occur in chronological order to facilitate a better reading and understanding of the project itself.

5.1.1 Project Management

We must remember the tasks that are developed in this GEP course in order to contrive this project. Some of the tasks handed in have helped to deal with the time to carry out the writing of the project. There are a total of five listed below:

- task 1: Defining the scope and contextualization (24.5 h)
- task 2: Timing (8.25 pm)
- task 3: Economic Management and Sustainability (9.25 pm)
- task 4: Preview (6.25 pm)
- task 5: Specification (8.5 h)

Each requires hours of dedication that makes a total sum of 56.75. To this we must also add the oral presentation and final document that needs an estimated 18.25 hours. Therefore the total amount of time would be about 75 hours.

In order to perform all these tasks, we need different resources listed below:

- Computer
- Microsoft Word (or any text editor)
- Adobe Reader (or reader of PDF's)
- FIB Platform Website
- Athena Web platform
- Microsoft Office Project
- Google Drive

5.1.2 Product Research

Once we had the idea of the product, the first step was to investigate whether it already existed. If so we needed to see how the product was executed and whether there are differences with the obtained idea. If not, see if there are things alike. This task, which may seem insignificant, is very important and the time needed to develop it is variable.

In case the product already existed, we need between 5 and 10 hours to see if it is a relatively new idea or it was put forward long ago. Also look who created it and where it was created. Then you need to do some research of the product sold and observe the differences with the idea that we thought. See also what technology they used and, finally, if their product could be improved or not.

In case the product does not exist, which would be very rare nowadays, it should be sought if there is a similar product or there had been other attempts to carry it out. In this case, time is increased due to the uncertainty of whether or not it exists, thus looking for similar products can take very long. Therefore, as timing is concerned, we could very well be talking of about15 to 20 hours.

To perform these tasks we simply need a computer with internet and look into specialized books. You also need some skills to properly search and save time.

5.1.3 Product Creation

This section has different subsections because of its complication. These subsections will be defined by chronological order.

5.1.3.1 Research Technologies

To make the chips in the form of stickers and make all this project work, various existing technologies must be evaluated. First they must be investigated and see if there is any new one to help us make the chips. Although at first it can be quite clear which technology to choose, you may not know all of them and, at the same time it is possible that a certain technology has not been considered and it happens to work perfectly.

This task as well as the research of the product can be variable. But, this time it will be based on where we are looking for the information and the knowledge we have of the current technologies. Therefore there is a 25 hours range.

We need a computer with Internet connection to carry out this task.

5.1.3.2 Choosing Hardware

The hardware that will make the chip completely has to be chosen correctly in order to create the device. We should check the characteristics of each component to see if they fit well.

Overall it is considered that about 20 or 25 hours will be required to finish this stage, also considering the time necessary to understand and become familiar with the resources to carry it out.

The resources required for this task are a computer with internet connection and hardware knowledge.

5.1.3.3 Configuration, development and installation of Hardware

Once we know what the components should be, we must start with the setting up and programming. It is important to say that every single task, no matter how insignificant, has its importance because that is the basis for the project to work properly. Among other functions, the chip will have to set the Adafruit Bluefruit up to act as a beacon. A GATT service must also be added with a feature to establish some communication between the beacon and the Smartphone. Finally, this feature will be constantly consulted to check whether there has been any interaction between the phone and the beacon.

To ensure a smooth operation we would require from 75 to 100 hours as long as there are no problems such as the inadequate functioning of the beacon or a factory error.

The resources required for this stage are Arduino software, the selected components, male cables - male connections for cable USB - micro USB connection to Arduino and of course a computer to program.

5.1.3.4 Hardware Welding

Once everything works correctly it will be necessary to weld the hardware to leave it as we wish. We must also say that a 3D box has been designed in this task in order to put the hardware and thus, have a more pleasant presentation.

This task may last in-between 50 / 75 hours, depending on whether the final result is as expected and everything works as planned.

The resources required for this task will be a soldering iron, tin, a connections plate and the components. In addition the design of the box was need from Sketchup software and a plugin from it.

5.1.3.5 Implementing the mobile app

The development of this application can be very variable as far as time is concerned. It is knowledgeable that during the development of applications like programming the hardware, the time varies depending on errors that we come across while programming.

It should also be mentioned that the program components, such as Bluetooth and other, complicate the accuracy of time due to the problems that may arise.

As for the programming of Bluetooth low energy, although there is extensive documentation online, familiarization is needed because of its different functions. It also must be said that the fact that one has never made android applications, has made the task even more complicated, since it is necessary to study and understand how it operates.

The time is estimated to be between 125 and 150 hours, which can be extended for another week due to the problems mentioned above.

The resources required for this program are the different components (Arduino Nano, speaker and Adafruit Bluefruit) in perfect operation, a computer, software for applications as Eclipse or even Unity and finally a mobile device to try the application.

5.1.4 Validity of results

Once it is believed that everything is working perfectly, we need time to make sure, and run various tests on it. Therefore an estimated 25 hours to perform them is necessary. These tests:

- The distance to the arriving Bluetooth data.
- Inconveniences than arise when putting the chip or other object in the bag.
- Operation of the speaker.
- Checking of the mobile energy consumption.
- Other.

The only resources required for this task are a fully operational chip and a mobile device with the mobile app installed.

5.1.5 Creating more products (optional)

Finally, when the product is fully operational and everything is working according to the expected results, then an optional task which is creating more products can be considered. This would allow us to see how the phone works with more than one device attached to it. For example: if the energy consumption soars or if more errors arise in the mobile application.

As everything would already be programmed and different circuits would be printed, the time of this task would be simply the time of installation, estimated at 5 hours, taking into account that no errors would arise.

The resources required for this are a PCB, Bluetooth devices, speakers, button batteries, soldering iron and tin.

5.2 Estimated time for the task

The estimated total amount of time is summarized on the following table:

Task	Time
Project Management	75 hours
- Deliverable 1	24.5 hours
- Deliverable 2	8.25 hours
- Deliverable 3	9.25 hours
- Deliverable 4	6.25 hours
- Deliverable 5	8.5 hours
Product Research	15 – 20 hours
Product Creation	Total:
- Technologies Research	25 hours
- Choice of Hardware	25 hours
- Configuration, development and installation of Hardware	100 hours
- Hardware Welding	75 hours
- Implementing the mobile application	150 hours
Validity of results	25 hours
Creating more products	5 hours/product

Figure 30: Task table and time needed

The overall operation is 475 hours in the worst case, if we calculate 4-5 hours a day, 5 days a week (excluding Saturday and Sunday), that would take 4.75 months to make one prototype. We must take into count that it can vary in terms of certain tasks mentioned above.

5.3 Explanation of the premises

The units are each of the tasks described above. Until one is not over the other one cannot begin because one task requires of the one before.

We can see a summary of these dependencies on the table below:

Task	Unit
Product management	-
Product Research	Product management
Technology Research	Product Research
Choice of Hardware	Technologies Research
Configuration, development and installation	Choice of Hardware

of the Hardware	
Setting mobile application	Configuration, development and installation of the Hardware Implementation
Validity of the result	Implementation of the mobile app.
Creation of more products	Validity of the result

Figure 31: Tasks Units

It must be emphasized that the welding of hardware task has a dependency on configuration, development and installation of hardware, but it generates no dependency, that is, once the configuration, development and installation of hardware are over, this can be done at any moment.

6. Financial management and sustainability

This section will explain both the direct and indirect costs. Direct costs are those that are directly related to the tasks described in the previous section. Indirect costs are those that are shared with other activities, in our case, it could be power consumption, water, etc ... ADSL

The formula that has been used to calculate depreciation is:

- The unit price divided by the useful life in months, and then multiplied by months that it will be used. In this project it is estimated that the resources will used during 5 months.

6.1 Direct costs

The following grid shows all direct costs used for this project and classified by the previously defined tasks.

	Units	Unit Price (€)	Useful life (years)	Estimated depreciation (€)	Price (€)
Project Management					
Laptop	1	1540	5	128.33	128.33
Unexpected (repairing)	1	0	_		0*
Microsoft Word license	1	100	3	13.8888888	13.9
Software Microsoft Office Project	1	0	5	0	0
Software Adobe Reader	1	0	5	0	0
Fib Website Platform	1	0	8	0	0
Atenea Website Platform	1	0	8	0	0
Google Drive	1	0	8	0	0
Internet Connection Fees	1	40	6	2.77777777	2.8
Product Creation					

Hardware Configuration ,development and Setting					
Adafruit Bluefruit UART Friend	1	15.49	5	1.29083333	1.3
Unpredicted (ill functioning) Risk 10%	1	15.49	_		1.55
Speaker	1	1,89	5	0,1575	0,16
Unpredicted (ill functioning) Risk 10%	1	1,89	_		0,19
Battery CR-P2	1	2	0.25	2	2
Arduino nano	1	5,87	0.25	5,87	5,87
Unpredicted (ill functioning) Risc 10%	1	5,87	_		0,59
Hardware Welding					
Welder + tin	1	20	5	1.66666666	1.67
Software Sketchup (trial)	1	0	0.08	0	0
Pluggin Sketchup	1	0	0.08	0	0
Implementing mobile app					
Software Android Studio	1	0	5	0	0
Nexus 4	1	300	5	15	15
Human Resources (Computer Engineer)	250 h	20€/h	-	-	5500
Human Resources' (Project Director)	150 h	22€/h			3300
More Unpredicted issues (15% del total)	1	3500	0,25		1336,53

Figure 32: Direct Costs

* Laptop or mouse repairing is free because they are products purchased on 09/26/2015 and therefore are guaranteed.

The total direct cost is equal to 10.510.37 € with 15% contingency already added.

6.2 Indirect Costs

As in the previous section, the following grill contains the indirect costs of the project

	Units	Unit Price (€)	Useful life (years)	Estimated depreciation (€)	Price (€)
Electricity	475 h	0,045	-	-	15,75
Laboratory Room ESSAI Fib	20 h	0	–	–	0

Figure 33: Indirect Costs

6.3 Total Costs

The following table shows the total costs of the project:

Direct Costs	10510.37 €
Indirect Costs	15.75 €
Contingencies (3%)	315.78 €
Total	10841.87€ €

Figure 34: Total Costs

We must take into account that this is the cost to make the application and a single product. Manufacturing more products would result in a higher total price, but not necessarily double.

In order to do so every new product should be bought:

- Arduino nano
- Adafruit Bluefruit UART LE friend
- Speaker
- Battery CR-P2
- Box.

The price has not been calculated because as a big quantity should be required, it would be necessary to speak with some distributor in order to get a special price.

6.4 Cost control

One of the biggest risks in IT projects is the dead line modification, that is, whether the time established will finally be right.

In this project, in order to comply with the fixed duration, a specific and periodical record of the hours invested is used. Since we use the Scrum methodology, the morning meeting will able us to see how the record and is going and have a daily track.

7. Social, economical and environmental Sustainability

In order to get the job done efficiently, it is necessary to make a sustainability study. This project will take it from three different perspectives: the social, economical and environmental. Each receives a grade based on the Socratic Method, which is non more than making and answering a series of questions.

The following table shows the three dimensions with their corresponding grades:

Dimension	Grade
Economical	8,9
Social	7,2
Environmental	7,0

Figure 35: Dimensions and correspondent grades.

Economic dimension

- Assessment of costs has been made in the previous section, both human and material resources

- As shown in the previous section, there are planned costs of upgrades and repairs

- Taking into account the current market cost, this product could be perfectly competitive, although the purchase of some essential devices such as module ATmega328 should be reduced.

- It could possibly be done in less time, but not with fewer resources.

- Each task has the right measure of time, and this is proportional to its importance.

Taking into account the response of each of the issues, the economic dimension has scored 8.9.

Social dimension

- In Catalonia there currently is a fairly delicate social and political situation. Catalan people are trying to leave the Spanish state due to the economic and social deficit that Catalonia has suffered, so relations not only with Spain but also with Europe are deteriorating. Although we currently do not know how this process, which Catalonia is trying to carry out democratically, will end. Therefore it is possible that opportunities to create a new product in Europe may be difficult.

On a social level, Catalonia is in a very unfavorable position due to the high unemployment rate (19.88% 2nd half 2015).

- It could only affect us in case they sold millions of products, consequently needing much more staff.

- If we take a look at the purchasing of a very similar product in France, we could have a positive answer, although we should see the acceptance of this product first in Catalonia and after in Spain.

- As the product is based on not losing relevant or important objects, I believe that it definitely improves the quality life consumers' life quality.

-It will vary by the fact that important objects will not be lost, plus people will stop

worrying about where they put them.
- I do not think any group of people could be wronged by this TFG.

This dimension has received a grade of 7.2 for the social impact that this may have and by the answers obtained in the test.

Environmental dimension

- This question has been answered in the section on planning.
- Basically, it would be exactly the same, provided we do not ask for the printed circuit board in a company where they do not have the ecological footprint.
- You could reuse virtually all the parts that have been used, except for the printed circuit.
- The creation of printed circuit uses copper element is quite polluting.
- Being an external company which uses copper, it has to be assessed. Initially the company is Spanish; so it must comply with the same laws. So we could ensure that minimum wages for the workers and at the same time complying with environmental Spanish laws.
- Copper is used but sparingly because the printed circuit board is of a small size, and also there would be few tracks.
-Both the Bluetooth device and the speaker could be reused and finally the same processor.

This dimension has a score of 7.0, due to the small pollution generated by the production process.

8. Bibliography

(Last consultation of all links held on 10.17.2015)

Scrum Methodology:

- https://ca.wikipedia.org/wiki/Scrum#Organitzaci.C3.B3
- https://ca.wikipedia.org/wiki/Scrum#Organitzaci.C3.B3

Wistiki:

- http://www.wistiki.com/
- http://www.finanzas.com/noticias/empresas/20150717/wistiki-adios-estres-buscar-3185628.html

Near Field Communication:

- https://es.wikipedia.org/wiki/Near_field_communication

Computer components:

- https://www.amazon.es/Aptotec-ATmega328P-development-DuPont-cable-compatible/dp/B014TE52RS/ref=sr_1_1?ie=UTF8&qid=1458556735&sr=8-1-spons&keywords=arduino+nano&psc=1 (Arduino Nano)
- https://www.adafruit.com/products/2479 (Adafruit Bluefruit LE UART Friend)

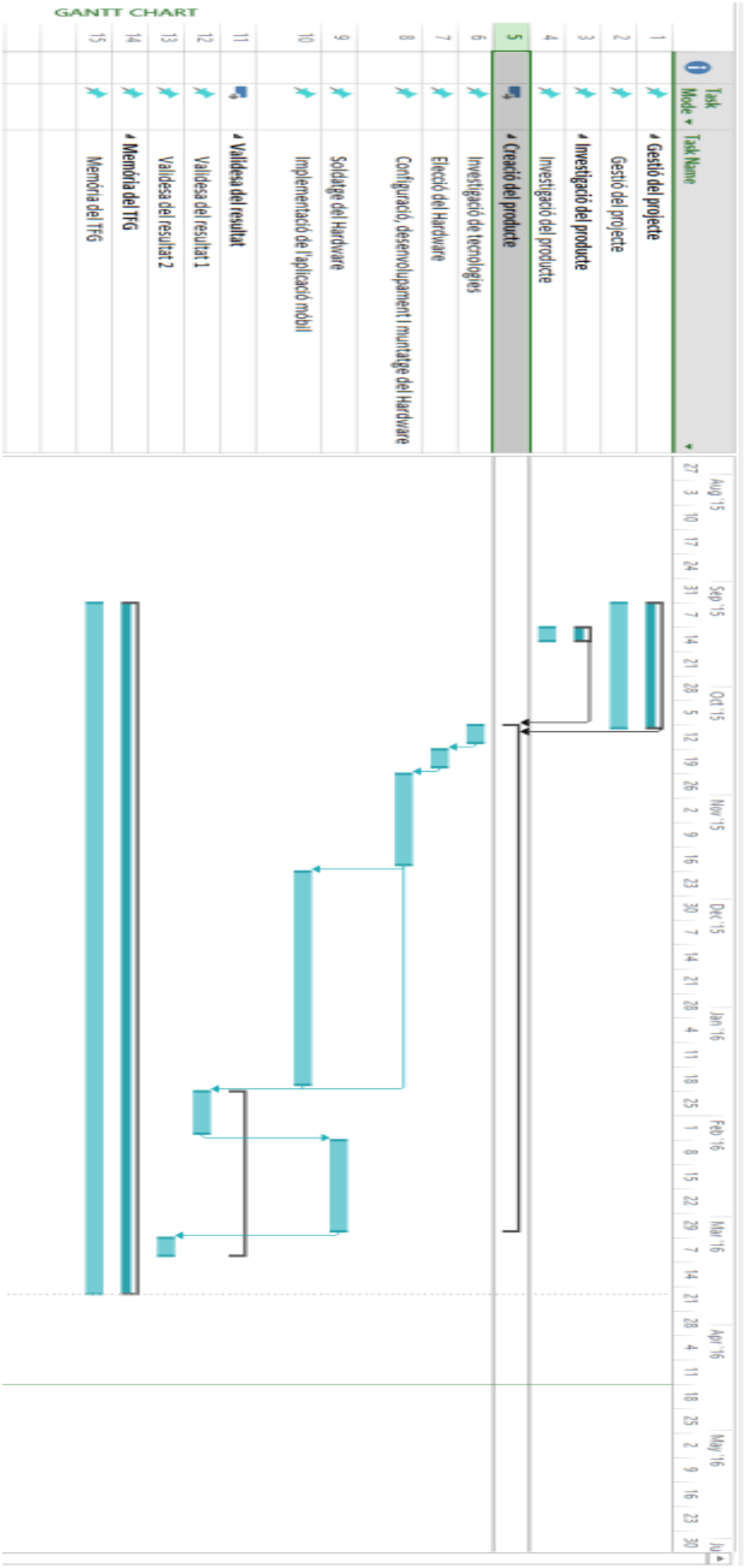
Other components:

- http://www.pccomponentes.com/unotec_kit_de_soldadura_completo.html (Welder)
- http://www.pccomponentes.com/kit_40_jumpers_de_20cm_macho_macho.html (Cables)
- http://www.diotronic.com/herramientas-instrumentacion/herramientas-y-equipos-de-taller/modulos-proto-board/protoboard-profesional-mini_r_936_3728.aspx(Protoboard)

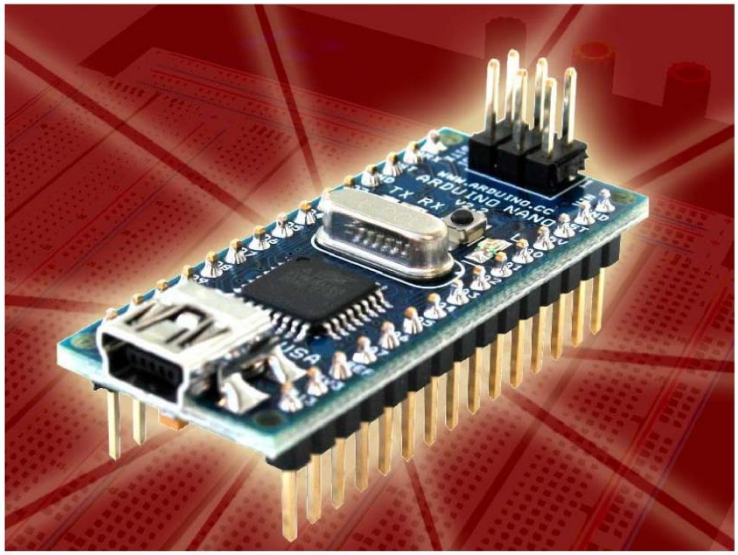
9. Index figures

10. ANNEX

Gantt Chart:



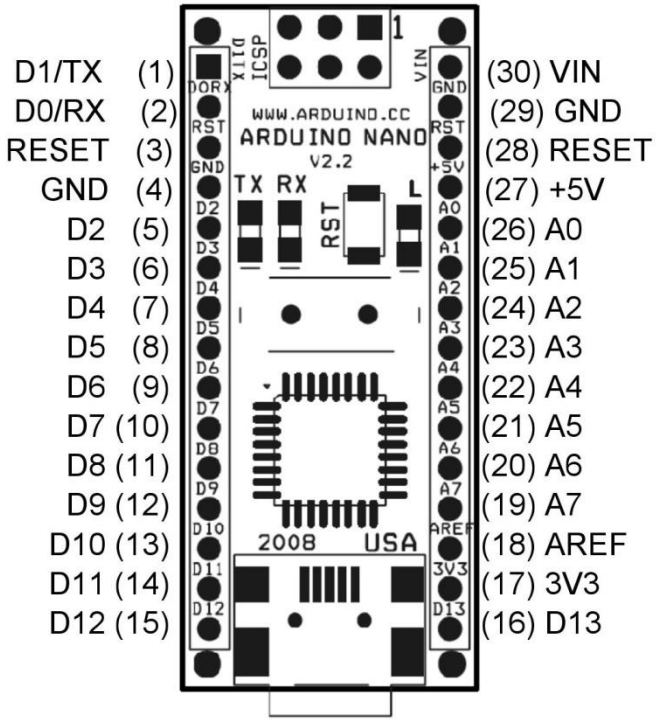
Annex: User Manual Arduino Nano



More information:

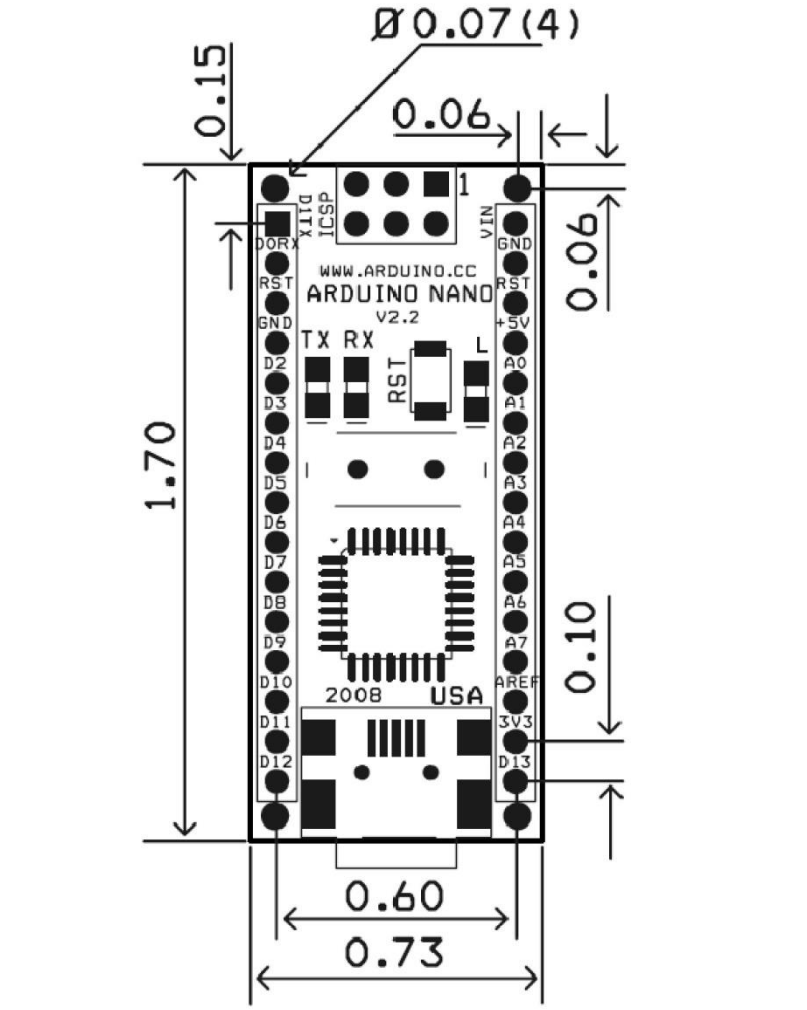
www.arduino.cc Rev. 2.3

Arduino Nano Pin Layout



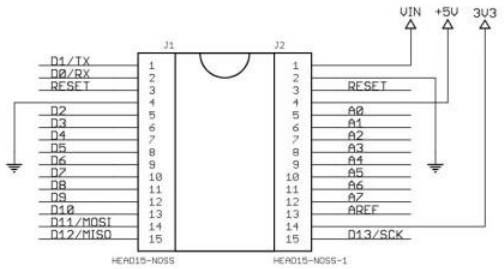
Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

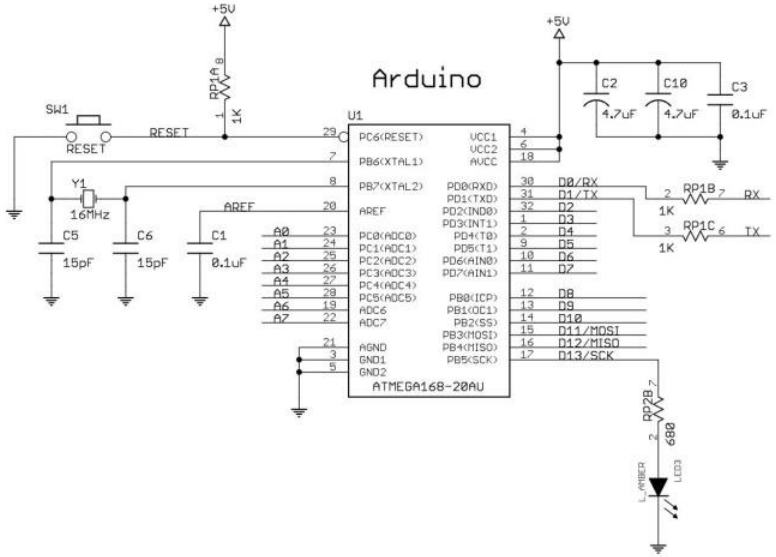
Arduino Nano Mechanical Drawing

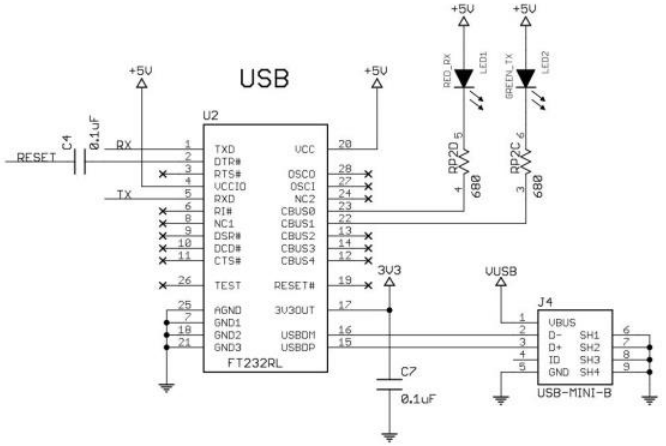


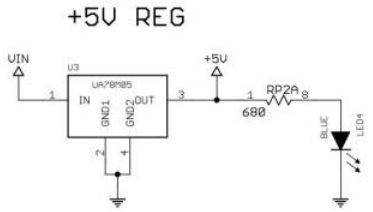
ALL DIMENTIONS ARE IN INCHES

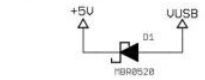
Arduino Nano Schematic

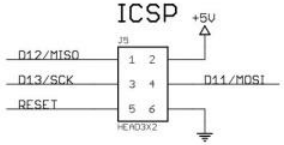


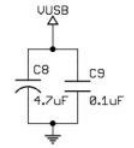












v2.3 - Modify FT232RL to use +5V	
TITLE: Arduino Nano	
Document Number:	REV: 2.3
Date: 6/26/2008 8:35:54 PM	Sheet: 1/1